Particle Filtering with Dynamic Shape Priors

Yogesh Rathi, Samuel Dambreville, Allen Tannenbaum *

Georgia Institute of Technology, Atlanta, GA,
{yogesh.rathi,samuel.dambreville,tannenba}@bme.gatech.edu

Abstract. Tracking deforming objects involves estimating the global motion of the object and its local deformations as functions of time. Tracking algorithms using Kalman filters or particle filters have been proposed for tracking such objects, but these have limitations due to the lack of dynamic shape information. In this paper, we propose a novel method based on employing a locally linear embedding in order to incorporate dynamic shape information into the particle filtering framework for tracking highly deformable objects in the presence of noise and clutter.

1 Introduction

The problem of tracking moving and deforming objects has been a topic of substantial research in the field of active vision; see [1, 2] and the references therein. There is also an extensive literature with various proposals for tracking objects with static shape prior [3]. This paper proposes a novel method to incorporate dynamic shape priors into the particle filtering framework for tracking highly deformable objects in the presence of noise and clutter.

In order to appreciate this methodology, we briefly review some previous related work. The possible parameterizations of planar shapes described as closed contours are of course very important. Various finite dimensional parameterizations of continuous curves have been proposed, perhaps most prominently the B-spline representation used for a "snake model" as in [2]. Isard and Blake (see [1] and the references therein) use the B-spline representation for contours of objects and propose the CONDENSATION algorithm [1] which treats the affine group parameters as the state vector, learns a prior dynamical model for them, and uses a particle filter [4] to estimate them from the (possibly) noisy observations. Since this approach only tracks affine parameters, it cannot handle local deformations of the deforming object.

Another approach for representing contours is via the level set method [5, 6] where the contour is represented as the zero level set of a higher dimensional

* This research was supported by grants from NSF, NIH (NAC P41 RR-13218 through Brigham and Women's Hospital), AFOSR, ARO, MURI, and MRI-HEL, and the Technion, Israel Insitute of Technology. This work was done under the auspices of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Particle Filtering with Dynamic Shape Priors	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Georgia Institute of Technology,School of Electrical and Computer Engineering,Atlanta,GA,30332**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**12**	

function, usually the signed distance function [5]. For segmenting an object, an initial guess of the contour (represented using the level set function) is deformed until it minimizes an image-based energy functional. Some previous work on tracking using level set methods is given in [3, 7–10].

Shape information is quite useful when tracking in clutter, especially if the object to be tracked gets occluded. Hence, a number of methods have been proposed [3] which incorporate a static shape prior into the tracking framework. The approach of these works is based on the idea that the object being tracked does not undergo a deformation (modulo a rigid transformation). Another method to obtain a shape prior is using PCA (principal component analysis) [11]. In this case, it is assumed that the shape can undergo small variations which can be captured by doing linear PCA. However, linear PCA is quite inadequate in representing the shape variations if the object being tracked undergoes large deformations (as will be explained in detail in the subsequent sections).

The authors in [12] use a particle filtering algorithm for geometric active contours to track highly deformable objects. The tracker however fails to maintain the shape of the object being tracked in case of occlusion. The present work extends the method proposed in [12] by incorporating dynamic shape priors into the particle filtering framework based on the use of a Locally Linear Embedding (LLE). LLE [13, 14] attempts to discover the nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions. To the best of our knowledge, this is the first time LLE has been used for shape analysis and tracking. Another approach closely related to our work was proposed in [15], wherein exemplars were used to learn the distribution of possible shapes. A different method in [16] separates the space of possible shapes into different clusters and learns a transition matrix to transition from one patch of shapes to the next. Our approach is different from those in [15, 16] in that we do not learn the dynamics of shape variation *apriori*. The only knowledge required in our method is a possible set of shapes of the deforming object.

The literature reviewed above is by no means exhaustive. Due to paucity of space we have only quoted a few related works. The rest of the paper is organized as follows: Section 2 gives the motivation and briefly describes the concepts of LLE, shape similarity measures, and curve evolution. Section 3 develops the state space model in detail and Section 4 describes the experiments conducted to test the proposed method. Some conclusions and further research directions are discussed in Section 5.

2 Preliminaries

Principal component analysis (PCA) is one of the most popular forms of dimensionality reduction techniques. In PCA, one computes the linear projections of greatest variance from the top eigenvectors of the data covariance matrix. Its first application to shape analysis [11] in the level set framework was accomplished by embedding a curve C as the zero level set of a signed distance function Φ. By doing this, a small set of coefficients can be utilized for a shape prior in various

segmentation tasks as shown in [11, 17]. However, linear PCA assumes that any required shape can be represented using a linear combination of eigen-shapes, i.e., any new shape $\tilde{\Phi}$ can be obtained by [17], $\tilde{\Phi} = \bar{\Phi} + \sum_{i=1}^{N} w_i \Phi_i$, where w_i are weights assigned to each eigenshape Φ_i and $\bar{\Phi}$ is the mean shape. Thus, PCA assumes that the set of training shapes lie on a linear manifold.

More specifically, let us consider shapes of certain objects with large deformations, for example, Figure 1 shows a set of few shapes of a man. PCA was performed on 75 such shapes (embedded in a signed distance function). Figure 2 shows the original and the reconstructed shape. Thus, linear PCA cannot be used to obtain a shape prior if the training set lies on a non-linear manifold.

Fig. 1. Few shapes of a man from a training set. Note the large deformation in shape.

Fig. 2. Left: Original shape, Middle: projection in the PCA basis, Right: LLE (2 nearest neighbors).

In [18], the authors proposed an unsupervised Locally Linear Embedding (LLE) algorithm that computes low dimensional, neighborhood preserving embeddings of high dimensional data. LLE attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear combinations. It has been used in many pattern recognition problems for classification. In this work, we use it in the particle filtering framework for providing dynamic shape prior.

2.1 Locally Linear Embedding for Shape Analysis

The LLE algorithm [14] is based on certain simple geometric intuitions. Suppose the data consists of N vectors Φ_i sampled from some smooth underlying manifold. Provided there is sufficient data, we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. In the simplest formulation of LLE, one identifies k nearest neighbors for each data point. Reconstruction error is then measured

by the cost function: $E(W) = \left(\Phi - \sum_j w_j \Phi_j\right)^2$. We seek to minimize the reconstruction error $E(W)$, subject to the constraint that the weights w_j that lie outside the neighborhood are zero and $\sum_j w_j = 1$. With these constraints, the weights for points in the neighborhood of Φ can obtained as [13]:

$$w_j = \frac{\sum_{m=1}^{k} R_{jm}}{\sum_{p=1}^{k}\sum_{q=1}^{k} R_{pq}}, \text{where } Q_{jm} = (\Phi - \Phi_j)^T(\Phi - \Phi_m), \quad R = Q^{-1} \tag{1}$$

In this work, we assume that a closed curve C_j is represented as the zero level set of a signed distance function Φ_j. Stacking all the columns of Φ_j one below the other, one can obtain a vector of dimension D^2, if Φ_i is of dimension $D \times D$. (In the rest of the paper, we use Φ interchangeably to represent a vector of dimension D^2 or a matrix of dimension $D \times D$. The appropriate dimension can be inferred from the context.) Figure 2 shows a particular shape being represented by 2 of its nearest neighbors.

2.2 Finding the Nearest Neighbors

The previous section showed how to represent a shape Φ_i by a linear combination of its k neighbors. Here we consider the key issue of how to find the nearest neighbors. One might be tempted to use the Euclidean 2-norm to find distance between shapes, i.e., if $d^2(\Phi_i, \Phi_j)$ is the (squared) distance between Φ_i and Φ_j, then $d^2(\Phi_i, \Phi_j) = \| \Phi_i - \Phi_j \|^2$. However, this norm does not represent distance between shapes, but only distance between two vectors. Since we are looking for the nearest neighbors of C_i in the shape space, a similarity measure between shapes is a more appropriate choice. Many measures of similarity have been reported; see [19, 3, 20]. In this paper, we have chosen the following distance measure [21]:

$$d^2(\Phi_i, \Phi_j) = \int_{p \in Z(\Phi_i)} EDT_{\Phi_j}(p)dp + \int_{p \in Z(\Phi_j)} EDT_{\Phi_i}(p)dp \tag{2}$$

where, EDT_{Φ_i} is the Euclidean distance function of the zero level set of Φ_i (one can think of it as the absolute value of Φ_i), and $Z(\Phi_i)$ is the zero level set of Φ_i. We chose this particular distance measure because it allows for partial shape matching which is quite useful for occlusion handling. More details about this measure may be found in [21]. We should note that the development of the remaining algorithm does not depend on the choice of the distance measure. Thus, once the distance measure between each Φ_i and the rest of the elements in the training set is known, one can find the nearest neighbors of Φ_i.

2.3 Curve Evolution

There is a large literature concerning the problem of separating an object from its background [3, 9]. Level sets have been used quite successfully for this task.

In [22], the authors have proposed a variational framework for segmenting an object using the first two moments (mean and variance) of image intensities. In the present work, we have used the energy functional given in [22]

$$\begin{aligned} E_{image} = & \int_{\Omega} \left(\log\sigma_u^2 + \frac{(I(x)-u)^2}{\sigma_u^2} \right) H(\Phi) dx \\ & + \int_{\Omega} \left(\log\sigma_v^2 + \frac{(I(x)-v)^2}{\sigma_v^2} \right) (1 - H(\Phi))\, dx + \nu \int_{\Omega} \parallel \nabla H(\Phi) \parallel dx, \end{aligned} \tag{3}$$

which upon minimization gives the following PDE:

$$\frac{\partial \Phi}{\partial t} = \delta_\epsilon(\Phi) \left(\nu \mathrm{div} \frac{\nabla \Phi}{\parallel \nabla \Phi \parallel} + \log \frac{\sigma_v^2}{\sigma_u^2} - \frac{(I(x)-u)^2}{\sigma_u^2} + \frac{(I(x)-v)^2}{\sigma_v^2} \right). \tag{4}$$

Here $I(x)$ is the image, u, v are the mean intensities inside and outside the curve C (corresponding to Φ) respectively, σ_u^2, σ_v^2 are the respective variances and $\delta_\epsilon(\Phi) = \frac{dH}{d\Phi}$ is the Dirac delta function and H is the Heaviside function as defined in [22]. Note that, one could use any type of curve evolution equation in the algorithm being proposed. We have made this particular choice because it is simple yet powerful in segmenting cluttered images.

3 The State Space Model

This section describes the state space model, the prediction model, and the importance sampling concept used within the particle filtering framework for tracking deformable objects. We will employ the basic theory of particle filtering here as described in [4].

Let S_t denote the state vector at time t. The state consists of parameters T that models the rigid (or affine) motion of the object (e.g., $T = [x\ y\ \theta]$ for Euclidean motion) and the curve C (embedded as the zero level set of Φ) which models the shape of the object, i.e., $S_t = [T_t\ \Phi_t]$. The observation is the image at time t, i.e., $Y_t = Image(t)$. Our goal is to recursively estimate the posterior distribution $p(S_t|Y_{1:t})$ given the prior $p(S_{t-1}|Y_{1:t-1})$. This involves a time update step and a measurement update step as described in the next section.

In general, it is quite difficult to obtain a model for predicting the position and shape of the deforming object. More specifically, in the current case, it is very difficult to obtain samples from the infinite dimensional space of closed curves (shapes). This problem can be solved using Bayesian *importance sampling* [23], described briefly below: Suppose $p(x)$ is a probability density from which it is difficult to draw samples (but for which $p(x)$ can be evaluated) and $q(x)$ is a density which is easy to sample from and has a heavier tail than $p(x)$ (i.e. there exists a bounded region R such that for all points outside R, $q(x) > p(x)$). $q(x)$ is known as the *proposal density* or the *importance density*. Let $x^i \sim q(x)$, $i = 1, ..., N$ be samples generated from $q(\cdot)$. Then, an approximation to $p(\cdot)$ is given by $p(x) \approx \sum_{i=1}^{N} \omega^i \delta(x - x^i)$, where $\omega^i \propto \frac{p(x^i)}{q(x^i)}$ is the

normalized weight of the i-th particle. So, if the samples, $S_t^{(i)}$, were drawn from an importance density, $q(S_t|S_{1:t-1}, Y_{1:t})$, and weighted by $\omega_t^{(i)} \propto \frac{p(S_t^{(i)}|Y_{1:t})}{q(S_t^{(i)}|S_{1:t-1}^{(i)}, Y_{1:t})}$, then $\sum_{i=1}^{N} \omega_t^{(i)} \delta(S_t^{(i)} - S_t)$ approximates $p(S_t|Y_{1:t})$. The choice of the importance density is a critical design issue for implementing a successful particle filter. As described in [24], the proposal distribution $q(\cdot)$ should be such that particles generated by it, lie in the regions of high observation likelihood. Another important requirement is that the variance of the weights ω^i should not increase over time. Various algorithms have been proposed [24] to achieve this objective. One way of doing this is to use an importance density which depends on the current observation. This idea has been used in many past works such as the unscented particle filter [25] where the proposal density is a Gaussian density with a mean that depends on the current observation. In this work, we propose a possible importance density function $q(S_t|S_{t-1}, Y_t)$ and show how to obtain samples from it. Note that, the space of closed curves from which we want to obtain samples is infinite dimensional.

3.1 Time Update

The prediction $\hat{S}_t$ at time t is given by: $\hat{S}_t = f_t(S_{t-1}, i_t, n_t)$ where n_t is random noise vector, i_t is any user defined input data (in our case, it is the set of training data) and f_t is possibly a nonlinear function. The problem of tracking deforming objects can be separated into two parts [8]:

1. Tracking the global rigid motion of the object;
2. Tracking local deformations in the shape of the object, which can be defined as any departure from rigidity.

Accordingly, we assume that the parameters that represent rigid motion T_t and the parameters that represent the shape Φ_t are independent. Thus, it is assumed that the shape of an object does not depend on its location in the image, but only on its previous shape and the location of an object in space does not depend on the previous shape. Hence, the prediction step consists of predicting the spatial position of the object using $\hat{T}_t = T_{t-1} + n_t^{(T)}$ where $n_t^{(T)}$ is random Gaussian noise vector with variance σ_T^2. The prediction for shape $\hat{\Phi}_t$ is obtained as follows:

$$\hat{\Phi}_t = p_0\Phi_{t-1} + p_1\Phi_{t-1}^{(N_1)} + p_2\Phi_{t-1}^{(N_2)} + ... + p_k\Phi_{t-1}^{(N_k)} \tag{5}$$

where $p_0, p_1, ...p_k$ are user defined weights such that $\sum_i p_i = 1$ and $\Phi_{t-1}^{(N_i)}, i = 1..k$ are the k nearest neighbors of Φ_{t-1}. The nearest neighbors are obtained as described in Section 2.2. A more generalized formulation of the prediction step above can be obtained by sampling the weights p_i from a known distribution (for example, an exponential distribution) to obtain a set of possible shapes and then choosing the predicted shape from this set, based on certain criteria.

We should note that, one of the main contributions of this paper is the formulation of a scheme that allows to dynamically predict the shape of the object

without learning the sequence in which they occur (unlike the methods in [15, 16]). Thus, the only knowledge required in this prediction step is a training set of shapes. In particular, one does not need to sample from an infinite-dimensional space of shapes (curves) but only from a set containing the linear combination of k nearest neighbors of Φ_{t-1}. This not only reduces the search space dramatically, but also allows to sample from a finite set of possible shapes. Once the latest observation Y_t is obtained, one can update the prediction based on this information as explained in the following section.

3.2 Measurement Update

At time t, for each particle i, generate samples as described in the prediction step in (5). Using the image at time t (Y_t), a rigid transformation is applied to each $\hat{\Phi}_t^{(i)}$ (in particular $\hat{C}_t^{(i)}$) by doing L_r iterations of gradient descent on the image energy E_{image} with respect to the rigid transformation parameters T. The curve is then deformed by doing a few (L_d) iterations of gradient descent ("curve evolution") on the energy, E, i.e., we generate

$$\tilde{\Phi}_t^{(i)} = f_R^{L_r}(\hat{\Phi}_t^{(i)}, Y_t), \qquad \Phi_t^{(i)} = f_{CE}^{L_d}(\tilde{\Phi}_t^{(i)}, Y_t) \tag{6}$$

where $f_R^{L_r}(\Phi, Y)$ is given by (for $j = 1, 2, ..., L_r$)

$$r^0 = r,\ r^j = r^{j-1} - \alpha^j \nabla_r E_{image}(r^{j-1}, \Phi, Y),\ \ T = r^{L_r},\ \ f_R^{L_r}(\Phi, Y) = T\Phi \tag{7}$$

and $f_{CE}^{L_d}(\mu, Y)$ is given by (for $j = 1, 2, ..., L_d$)

$$\mu^0 = \mu,\ \ \mu^j = \mu^{j-1} - \alpha^j \nabla_\mu E(\mu^{j-1}, Y),\ \ f_{CE}^{L_d}(\mu, Y) = \mu^{L_d} \tag{8}$$

where $E = E_{image} + \beta E_{shape}$. The energy E_{image} is as defined in equation (3) and E_{shape} is defined by [3]: $E_{shape}(\Phi) = \int_\Omega \bar{\Phi}(x)dx$, where $\bar{\Phi}(x)$ is the contour obtained from a linear combination of the nearest neighbors of Φ, with weights obtained using LLE from equation (1). The corresponding curve evolution equation is given by

$$\frac{\partial \Phi}{\partial t} = \bar{\Phi}(x) \parallel \nabla \Phi \parallel . \tag{9}$$

This PDE tries to drive the current contour shape Φ towards the space of possible shapes and equation (8) tries to drive the current contour towards the minimizer of energy E which depends on the image and shape information. The parameter β is user defined and weights the shape information with the image information. The use of LLE to provide shape information for contour evolution is another main contribution of this paper.

Details about equation (7) can be obtained from [17]; and equation (8) may be implemented by summing the PDE's (4) and (9). We perform only L (L_d or L_r) iterations of gradient descent since we do not want to evolve the curve until it reaches a minimizer of the energy, E_{image} (or E). Evolving to the local minimizer is not desirable since the minimizer would be independent of

all starting contours in its domain of attraction and would only depend on the observation, Y_t. Thus the state at time t would loose its dependence on the state at time $t-1$ and this may cause loss of track in cases where the observation is bad. In effect, choosing L to be too large can move all the samples too close to the current observation, while a small L may not move the particles towards the desired region. The choice of L depends on how much one trusts the system model versus the obtained measurements. Note that, L will of course also depend on the step-size of the gradient descent algorithm as well as the type of PDE used in the curve evolution equation.

For each i, the sample $S_t^{(i)}$ thus obtained is drawn from the importance density $q(S_t^{(i)}|S_{t-1}^{(i)}, Y_t) = \mathcal{N}(S_t^{(i)}, \Sigma)$, where we assume a Gaussian fit for the density $q(.)$ centered at each $S_t^{(i)}$. We further assume that the variance Σ is very small and constant for all particles, i.e., $q(S_t^{(i)}|S_{t-1}^{(i)}, Y_t) = constant$. We should note that, this methodology, even though sub-optimal (to the best of our knowledge, an optimal method to sample from an infinite dimensional space of curves does not exist) allows to obtain samples that lie in region of high likelihood. The above mentioned step of doing gradient descent can also be interpreted as an MCMC move step, where particles are "moved" to region of high likelihood by any available means, as given in [24].

3.3 Setting the Importance Weights

In this paper, the state process is assumed to be Markov, and the observations are conditionally independent given the current state i.e., $p(Y_t|S_{0:t}) = p(Y_t|S_t)$. This gives the following recursion for the weights [23]: $\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(Y_t|S_t^{(i)})p(S_t^{(i)}|S_{t-1}^{(i)})}{q(S_t^{(i)}|S_{t-1}^{(i)}, Y_t)}$. The probability $p(Y_t|S_t)$ is defined as $p(Y_t|S_t) \propto e^{\frac{-E(S_t,Y_t)}{\sigma_{Tot}^2}}$. We define $p(S_t|S_{t-1}) = p(T_t|T_{t-1})\ p(\Phi_t|\Phi_{t-1})$ with

$$p(T_t|T_{t-1}) \propto e^{\frac{-|T_t - T_{t-1}|}{\sigma_T^2}},\ p(\Phi_t|\Phi_{t-1}) \propto e^{\frac{-d^2(\Phi_t,\Phi_{t-1})}{\sigma_d^2}} + a\, e^{\frac{-d^2(\Phi_t,\breve{\Phi}_{t-1})}{\sigma_d^2}} \tag{10}$$

where d^2 is the (squared) distance measure defined above in (2), and $\breve{\Phi}_{t-1}$ is the MAP (maximum a-posteriori) estimate of the shape at time $t-1$. We should note that, using the MAP shape information available from time $t-1$ is quite essential, since it adds weights to particles which are closer to the previous best estimate than particles that are far away. This is quite useful in case of occlusion wherein particles which look like the previous best shape are given higher probability, despite the occlusion. The parameter a is user defined.

Based on the discussion above, the particle filtering algorithm can be written as follows:

- Use equation (5) to obtain $\hat{T}_t, \hat{\Phi}_t$.
- Perform L_r steps of gradient descent on rigid parameters using (7) and L_d iterations of curve evolution using (8).

- Calculate the importance weights, normalize and resample [4], i.e.,

$$\tilde{\omega}_t^{(i)} \propto p(Y_t|S_t^{(i)})p(T_t^{(i)}|T_{t-1}^{(i)})p(\Phi_t^{(i)}|\Phi_{t-1}^{(i)}),\ \omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^{N}\tilde{\omega}_t^{(j)}}\ . \tag{11}$$

4 Experiments

The proposed algorithm was tested on 3 different sequences and the results are presented in this section. We certainly do not claim that the method proposed in this paper is the best one for every image sequence on which it was tested, but it did give very good results with a small number of particles on all of the image sequences. We should add that to the best of our knowledge this is the first time dynamic shape prior in a level set framework has been used in conjunction with the particle filter [4] for tracking such deforming objects.

In all of the test sequences, we have used the following parameters which gave good results:

1. Choosing k, the number of nearest neighbors (for obtaining $\bar{\Phi}(x)$ in eqn (9)): k will depend on the number of similar shapes available in the training set [18]. In our experiments, $k = 2$ gave acceptable results.
2. Choosing σ_d^2: A classical choice [20] is $\sigma_d^2 = c\ \frac{1}{N}\sum_{i=1}^{N}\min_{j\neq i} d^2(\Phi_i, \Phi_j)$. For all the test sequences, $c = 1/20$ was used.
3. σ_T^2 models the motion dynamics of the object being tracked. In all the test sequences, since the spatial motion of the object was not large, we used $\sigma_T^2 = 1000$. Also, only translational motion was assumed, i.e., $T = [x\ y]$.

4.1 Shark Sequence

This sequence has very low contrast (object boundaries in some images are barely visible even to human observers) with a shark moving amid a lot of other fish which partially occlude it simultaneously in many places. This results in a dramatic change in the image statistics of the shark if a fish from the background occludes the shark. The training set was obtained by hand segmenting 10% of the images from the image sequence. Tracking results without shape information using the algorithm in [12] is shown in Figure 3. As can be seen, even though the algorithm tracks the shark, it is unable to maintain the shape. Results using the proposed algorithm are shown in Figure 3. This sequence demonstrates the robustness of the proposed algorithm in the presence of noise and clutter. The following parameters were used in tracking this sequence: $L_r = 1$, $L_d = 10$, particles = 40.

4.2 Octopus Sequence:

As seen in Figure 4, the shape of the octopus undergoes large changes as it moves in a cluttered environment. It gets occluded for several frames by a fish

having the same mean intensity. Tracking this sequence using equation (4) or any other method without shape information may result in the curve leaking to encompass the fish. Figure 4 shows tracking results using the proposed method. The following set of parameters were used in tracking this sequence: $L_r = 3$, $L_d = 10$, particles = 50, training set included 9% of possible shapes.

4.3 Soccer Sequence:

This sequence tracks a man playing soccer. There is large deformation in the shape due to movement of the limbs (hands and legs) as the person tosses the ball around. The deformation is also great from one frame to next when the legs occlude each other and separate out. There is clutter in the background which would cause leaks if geometric active contours or the particle filtering algorithm given in [12] were used to track this sequence (see Figure 5). Results of tracking using the proposed method are shown in Figure 5. The following set of parameters were used to track this sequence: $L_r = 5$, $L_d = 18$, particles = 50, and 20% of the possible shapes were included in the training set (see Figure 1).

5 Conclusions and Limitations

In this paper, we have presented a novel method which incorporates dynamic shape prior information into a particle filtering algorithm for tracking highly deformable objects in presence of noise and clutter. The shape prior information is obtained using Locally Linear Embedding (LLE) for shapes. No motion or shape dynamics are required to be known for tracking complex sequences, i.e., no learning is required. The only information needed is a set of shapes that can appropriately represent the various deformations of the object being tracked.

Nevertheless, the current algorithm has certain limitations. First, it is computationally very expensive, as each particle has to be evolved for many iterations. Second, the training set should contain sufficient number of possible shapes of the object being tracked so that LLE can be used.

References

1. Blake, A., Isard, M., eds.: Active Contours. Springer (1998)
2. Terzopoulos, D., Szeliski, R.: Tracking with Kalman Snakes. In: Active Vision. MIT Press (1992) 3–20
3. Zhang, T., Freedman, D.: Tracking objects using density matching and shape priors. In: Proceedings of the Ninth IEEE International Conference on Computer Vision. (2003) 1950–1954
4. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/nongaussian bayesian state estimation. IEE Proceedings-F (Radar and Signal Processing) (1993) 140(2):107–113
5. Sethian, J.A.: Level Set Methods and Fast Marching Methods. 2nd edn. Cambridge University Press (1999)

6. Osher, S.J., Sethian, J.A.: Fronts propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of Computational Physics **79** (1988) 12–49
7. Paragios, N., Deriche, R.: Geodesic active contorus and level sets for the detection and tracking of moving objects. Transactions on Pattern analysis and Machine Intelligence **22**(3) (2000) 266–280
8. Yezzi, A., Soatto, S.: Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. Internaitonal Journal of Computer Vision **53**(2) (2003) 153–167
9. Dambreville, S., Rathi, Y., Tannenbaum, A.: Shape-based approach to robust image segmentation using kernel pca. In: IEEE, CVPR. (2006)
10. Dambreville, S., Rathi, Y., Tannenbaum, A.: Tracking deforming objects with unscented kalman filtering and geometric active contours. In: American Control Conference. (2006)
11. Leventon, M., Grimson, W.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: Proc. CVPR. (2000) 1316–1324
12. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A.: Particle filtering for geometric active contours with application to tracking moving and deforming objects. In: Proc. CVPR. (2005)
13. D.Ridder, Duin, R.: Locally linear embedding for classification. Technical Report PH-2002-01, Pattern Recognition Group, Delft University of Technology (2002)
14. Saul, L.K., Roweis, S.: (An introduction to locally linear embedding)
15. Toyama, K., Blake, A.: Probabilistic tracking in a metric space. In: ICCV. (2001) 50–59
16. Heap, T., Hogg, D.: Wormholes in shape space: Tracking through discontinuous changes in shape. In: ICCV. (1998) 344
17. Tsai, A., Yezzi, A., Wells, W., Tempany, C., et. al.: A shape based approach to the segmentation of medical imagery using level sets. In: IEEE Tran. On Medical Imaging. Volume 22. (2003)
18. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500) (2000) 2323–2326
19. Cremers, D., Soatto, S.: A pseudo-distance for shape priors in level set segmentation. In: IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision. (2003)
20. D. Cremers, S. Osher, S.S.: Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. Pattern Recognition,Tbingen, Springer LNCS **3175**(3) (2004) 36–44
21. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. In: ACM Transactions on Graphics (SIGGRAPH 2004). (2004)
22. Rousson, M., Deriche, R.: A variational framework for active and adaptative segmentation of vector valued images. Motion, Workshop 2002 **00** (2002) 56
23. Doucet, A., deFreitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer (2001)
24. Doucet, A.: On sequential monte carlo sampling methods for bayesian filtering. In: Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering. (1998)
25. van der Merwe, R., de Freitas, N., Doucet, A., Wan, E.: The unscented particle filter. In: Advances in Neural Information Processing Systems 13. (2001)

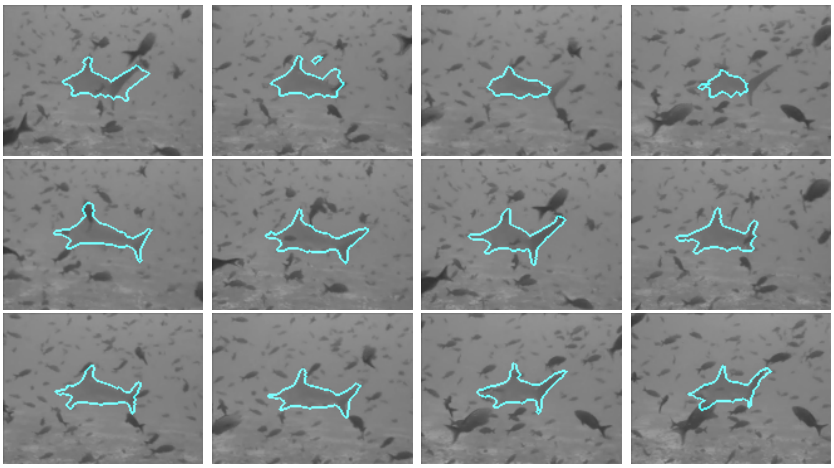

Fig. 3. First row shows tracking results with no shape information. Next two rows show results using the proposed algorithm.

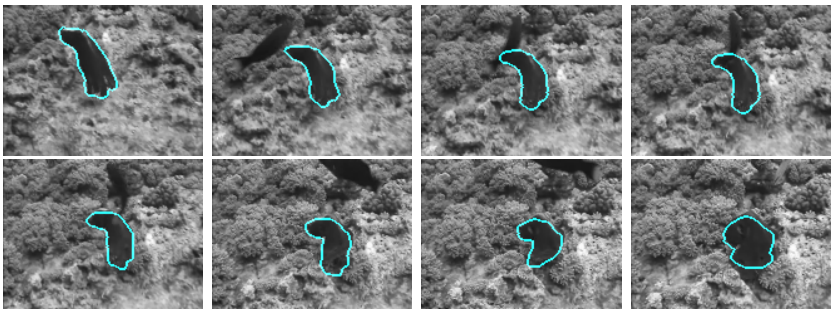

Fig. 4. Octopus Sequence: Results using the proposed algorithm. Notice that a fish with the same mean intensity occludes the octopus.

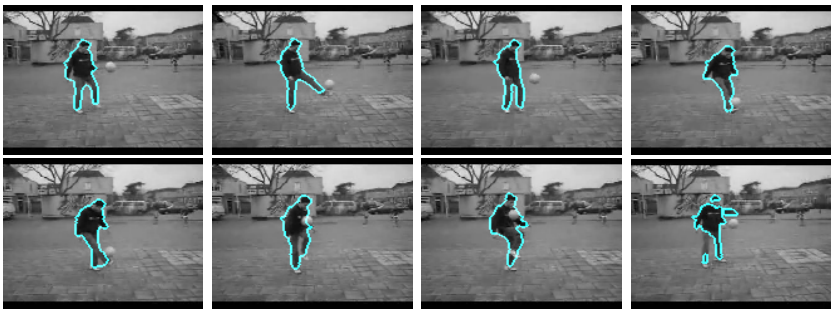

Fig. 5. Results of tracking using the proposed method. Last image at the bottom right is the segmentation using equation (4) without any shape information.